Hanguang 800 NPU
– The Ultimate AI Inference Solution for Data Centers

Yang Jiao, Liang Han, Xin Long, & Team

Alibaba Group
A Business-Driven Design

**TCO Efficient**
- For data center inference acceleration
- High throughput, low latency, power efficient
- High effective TOPs

**CNN Optimized**
- Convolution-efficient architecture
- As well as GEMM acceleration

**Domain Programmable**
- Native OPs for computer vision deep learning tasks
- Expand to support future activation functions
Optimized for CV Tasks

- Classification
- Object Detection
- Segmentation
- Point Cloud
Top Level Diagram

- **Top Level:**
  - 4 Cores w/ a ring bus
  - Command Processor (CP)
  - PCIE gen4 X16

- **Each Core:**
  - Tensor Engine (TE)
  - Pooling Engine (PE)
  - Memory Engine (ME)

- **SRAM-Only:**
  - 192MB Local Memory (LM)
  - Distributed shared
  - No DDR
Tensor Engine

- **Weight+Activation Stationary**
  - Data reuse
    - via W-buffer, A-buffer, Multicasting
  - Avoid img2col()
    - via sliding-window access
- **Fused OPs**
  - e.g. CONV-BN-ResidualADD-ReLU
Pooling Engine

- **POOL Unit**
  - POOLs, ROIs
  - ROI_align_2
- **INTP Unit**
  - Interpolations
  - ROI_align_1
- **Scale/Bias Unit**
  - Scaling/Bias operations
  - Data formatting

Diagram:
- From LM
- Int
- Pooling, ROI
- Interpolation
- ROI Align
Memory Engine

Ring Bus

LM

ME

2Kb

2Kb

PCIE_DST

CPY_SRC

CPY_DST

TRANS_SRC

TRANS_DST

MATRIX_SRC

WEIGHT_DST

FLATTEN_SRC

FALTTEN_DST

Credit memory

Tensor copy

Matrix transpose

Matrix to weight

Image flatten

PCIE & x-core

Reshape memory

Matrix to weight

Image flatten

Ring Bus

Matrix

weights

Ring Bus

matrix

weights
192MB On-Chip SRAM

- 1-R/W SRAM Modules
  - High density, low power
- 2 blocks for W & A
  - 16 modules in each block
- 4 clusters
  - Distributed shared
- 48MB in each of the 4 cores
  - Memory copy across cores via ME & Ring Bus

LM Organization in Each Core
Compressed and Quantized Storage/Processing

- **Compressed** Model
  - Sparsity Engine to unpack with bit-masks
  - Pruning is optional though
- **Quantized** computation and storage
- Vector Unit w/ **FP-24**
  - 1sign.8exp.15man
Workflow at Command Level
Workflow at Instruction Level

- Domain-specific instruction set
  - CISC-like, operation-fusion
  - Coarse-grain data at tensor level (HW decomposes & ctrl)

- Synchronization among 3 engines
  - Embedded bits in instructions
  - Hardware dependency checker
Scalable Task Mapping

For Small/Medium Models
- Model
  - Model-0
  - Model-1
  - Model-2
  - Copy-0
  - Copy-1
  - Copy-2
  - Copy-3
- Core-0
  - Core-1
  - Core-2
  - Core-3

Data Parallelism
Multiple Models
Layer Pipelining
Model Parallelism
Hybrid Parallelism
Multi-Chip Pipelining

For Large Models
- Layers-0
  - Layers-1
  - Layers-2
  - Layers-3
- Weights-0
  - Weights-1
  - Weights-2
  - Weights-3
- CONVs
  - FC
- Core-0
  - Core-1
  - Core-2
  - Core-3

For X-Large Models
- Layers-0
  - Layers-1
  - Layers-2
  - Layers-3
  - Layers-n
- Chip-0
  - Chip-1
  - Chip-n
- PCIE Switch

For Small/Medium Models
- Data Parallelism
- Multiple Models
- Layer Pipelining
- Model Parallelism

For Large Models
- Data Parallelism
- Multiple Models
- Layer Pipelining
- Model Parallelism
- Hybrid Parallelism
- Multi-Chip Pipelining
### Implementation

<table>
<thead>
<tr>
<th></th>
<th>NVIDIA T4</th>
<th>NVIDIA V100</th>
<th>Huawei Ascent910</th>
<th>NVIDIA A100</th>
<th>Alibaba Hanguang 800</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Peak Performance</strong> (INT8 TOPS)</td>
<td>130</td>
<td>125 (FP16)</td>
<td>512</td>
<td>624/1248</td>
<td>825</td>
</tr>
<tr>
<td><strong>Frequency</strong> (MHz)</td>
<td>1590</td>
<td>1530</td>
<td>1000</td>
<td>1410</td>
<td>700</td>
</tr>
<tr>
<td><strong>TDP</strong> (Watt)</td>
<td>70</td>
<td>300</td>
<td>350</td>
<td>400</td>
<td>280</td>
</tr>
<tr>
<td><strong>Area</strong> (mm^2)</td>
<td>545</td>
<td>815</td>
<td>1228</td>
<td>826</td>
<td>709</td>
</tr>
<tr>
<td><strong>Process</strong> (nm)</td>
<td>TSMC 12nm</td>
<td>TSMC 12nm</td>
<td>TSMC 7+nm</td>
<td>TSMC 7nm</td>
<td>TSMC 12nm</td>
</tr>
</tbody>
</table>

---

**Note:** The diagrams and images provide visual representations of the implementation and hardware components.
Software Stack

Frameworks

Compiler

Run-time

Application Infer Runtime

CPU Operators

NPUEngine Operators

RunTime

User Mode Driver

Kernel Mode Driver

CPU

NPU
Quantization Method

- Post-training quantization
  - Pruning is optional
- Static quantization at compile time
- Clip out-of-range post-rounding values

Symmetric quantization

- Weights: per-channel
- Activations: per-tensor
## Quantization Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>GPU FP32</th>
<th>NPU INT8</th>
</tr>
</thead>
<tbody>
<tr>
<td>resnet_v1_50 (Top1/Top5)</td>
<td>75.18 / 92.18</td>
<td>74.93 / 92.47</td>
</tr>
<tr>
<td>resnet_v1_101 (Top1/Top5)</td>
<td>76.40 / 92.90</td>
<td>76.02 / 93.04</td>
</tr>
<tr>
<td>resnet_v1_152 (Top1/Top5)</td>
<td>76.80 / 93.20</td>
<td>76.45 / 93.41</td>
</tr>
<tr>
<td>resnet_v2_50 (Top1/Top5)</td>
<td>75.57 / 92.82</td>
<td>75.11 / 92.92</td>
</tr>
<tr>
<td>resnet_v2_101 (Top1/Top5)</td>
<td>77.00 / 93.70</td>
<td>76.58 / 93.81</td>
</tr>
<tr>
<td>inception_resnet_v2 (Top1/Top5)</td>
<td>80.40 / 95.30</td>
<td>80.19 / 95.26</td>
</tr>
<tr>
<td>inception_v3 (Top1/Top5)</td>
<td>77.99 / 93.94</td>
<td>77.86 / 94.07</td>
</tr>
<tr>
<td>inception_v4 (Top1/Top5)</td>
<td>80.2 / 95.2</td>
<td>80.11 / 95.20</td>
</tr>
<tr>
<td>ssd_resnet_50_fpn (mAP)</td>
<td>37.12</td>
<td>36.96</td>
</tr>
<tr>
<td>ssd_inception_v2_coco (mAP)</td>
<td>24</td>
<td>26.27</td>
</tr>
<tr>
<td>faster_rcnn_resnet50_coco (mAP)</td>
<td>25.89</td>
<td>25.96</td>
</tr>
<tr>
<td>faster_rcnn_inception_v2_coco (mAP)</td>
<td>24.8</td>
<td>24.66</td>
</tr>
<tr>
<td>mask_rcnn_resnet50_atrous_512 (mAP)</td>
<td>20.77</td>
<td>20.8</td>
</tr>
</tbody>
</table>
Performance & Efficiency

ResNet-50 v1 (224x224) inference at INT8

Throughput Performance (Normalized)
Power Efficiency (Normalized)
Scalable Performance and Power

• Configurable frequency and voltage
• Can be deployed from data center to large edges

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Performance Mode</th>
<th>Efficiency Mode</th>
<th>Low Power Mode 1</th>
<th>Low Power Mode 2</th>
<th>Low Power Mode 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency (MHz)</td>
<td>700</td>
<td>475</td>
<td>475</td>
<td>475</td>
<td>475</td>
</tr>
<tr>
<td>Perf (img/sec)</td>
<td>78,563</td>
<td>53,983</td>
<td>37,500</td>
<td>25,000</td>
<td>12,500</td>
</tr>
<tr>
<td>Power (watt)</td>
<td>276</td>
<td>108</td>
<td>75</td>
<td>50</td>
<td>25</td>
</tr>
</tbody>
</table>
Batching-Independent Performance

Best in class latency, no performance drop

ResNet-50 v1 (224x224) inference at INT8
More Results

INT8 Inference Performance

<table>
<thead>
<tr>
<th>Resnet-101 V1</th>
<th>Habana.Goya</th>
<th>0.4ms</th>
<th>0.9ms</th>
<th>1.3ms</th>
<th>1.5ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.2ms</td>
<td>0.2ms</td>
<td>0.2ms</td>
<td>0.2ms</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inception V3</th>
<th>Habana.Goya</th>
<th>0.6ms</th>
<th>1.3ms</th>
<th>2.3ms</th>
<th>2.8ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.2ms</td>
<td>0.2ms</td>
<td>0.2ms</td>
<td>0.2ms</td>
<td></td>
</tr>
</tbody>
</table>

Latency (ms)

Throughput (img/sec)

Throughput (img/sec) vs Latency (ms) chart for Resnet-101 V1 and Inception V3.
MLPerf Inference V0.5 (Single-Chip Performance*)

ResNet-50 v1.5

<table>
<thead>
<tr>
<th>Offline Mode (img/s)</th>
<th>Stream Mode (Latency in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA T4 (Inf-0.5-2S)</td>
<td>NVIDIA T4</td>
</tr>
<tr>
<td>TPUC/ (Inf-0.5-1S)</td>
<td>TRU-3</td>
</tr>
<tr>
<td>Goya</td>
<td>NVIDIA Xavier</td>
</tr>
<tr>
<td>Intel Xeon 9262 Processor</td>
<td>Intel Xeon 8255C CPU</td>
</tr>
<tr>
<td>NVIDIA TITAN RTX</td>
<td>Intel® X86 Nervana + NNP</td>
</tr>
<tr>
<td>5680</td>
<td>69307</td>
</tr>
<tr>
<td>8203</td>
<td>1.37</td>
</tr>
<tr>
<td>14451</td>
<td>2.04</td>
</tr>
<tr>
<td>2983</td>
<td>69307</td>
</tr>
<tr>
<td>16563</td>
<td>69307</td>
</tr>
<tr>
<td>2159</td>
<td>69307</td>
</tr>
<tr>
<td>1292</td>
<td>69307</td>
</tr>
<tr>
<td>1218</td>
<td>69307</td>
</tr>
<tr>
<td>5284</td>
<td>69307</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server Mode (img/s)</th>
<th>MultiS Mode (img/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA T4 (Inf-0.5-2S)</td>
<td>NVIDIA T4</td>
</tr>
<tr>
<td>TPUC/ (Inf-0.5-1S)</td>
<td>TRU-3</td>
</tr>
<tr>
<td>Goya</td>
<td>NVIDIA Xavier</td>
</tr>
<tr>
<td>Intel Xeon 9262 Processor</td>
<td>Intel Xeon 8255C CPU</td>
</tr>
<tr>
<td>NVIDIA TITAN RTX</td>
<td>Intel® X86 Nervana + NNP</td>
</tr>
<tr>
<td>5193</td>
<td>45169</td>
</tr>
<tr>
<td>5258</td>
<td>45169</td>
</tr>
<tr>
<td>15008</td>
<td>45169</td>
</tr>
<tr>
<td>2425</td>
<td>45169</td>
</tr>
<tr>
<td>1005</td>
<td>45169</td>
</tr>
<tr>
<td>5131</td>
<td>45169</td>
</tr>
</tbody>
</table>

* Single-Chip Performance is not the primary metric of MLPerf. MLPerf name and logo are registered trademarks. See www.mlperf.org for more information.
Comparing with Latest GPUs

ResNet-50 v1.5 Inference

<table>
<thead>
<tr>
<th>GPU Model</th>
<th>Throughput Performance (Img/Sec)</th>
<th>Power Efficiency (Img/Sec/Watt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia.T4 (batch=128,int8)</td>
<td>5,097</td>
<td></td>
</tr>
<tr>
<td>Nvidia.V100 (DGX-1) (batch=128,Mixed)</td>
<td>7,097</td>
<td>24</td>
</tr>
<tr>
<td>Nvidia.A100 (batch=128,int8)</td>
<td>23,973</td>
<td>74</td>
</tr>
<tr>
<td>Alibaba.Hanguang800 (batch=1,int8)</td>
<td>69,307</td>
<td>248</td>
</tr>
</tbody>
</table>

Nvidia GPU Data Source: NVIDIA Data Center Deep Learning Product Performance
Deployment

In Data Centers

Smart City

Image Search

Smart Fashion

5G Edge Cloud

Smart Medical

Video Cloud

Recommender Sys

Smart Manufacture

Smart Retail

In Large End-Devices

Alibaba AI Cloud

In Edge Servers

Hanguang NPU Inside
Elastic Compute Service in Alibaba Cloud

X-Dragon architecture provides unified resource pool for Virtual Machines, Bare-Metals, and Containers.

Further Reading: “Introducing the Sixth Generation of Alibaba Cloud's Elastic Compute Service”
Elastic Bare-Metal AI Inference Instance

Public Cloud: ecs.ebman1.24xlarge
- CPU: Cascade Xeon 104 cores
- Memory: 384GB
- NPU: 4x 2-core Hanguang 800

Private Cloud
- Customized Configurations
- 1/2/4x 2-core/3-core/4-core Hanguang 800

X-Dragon Bare-metal Server
- Bare-metal performance
- Elasticity of Cloud Infrastructure

Support Common Framework
- Tensorflow
- MXNet
- PyTorch

Diagram:
- ECS or ECS Bare Metal Instance
- CPU
- Memory
- VirtIO-net
- VirtIO-blk
- External Devices
- VPC/SLB
- EBS Disk
- X-Dragon Hypervisor
- X-Dragon NiC
Deployed Applications via Alibaba Cloud

Top 6 businesses representing 99.1% deployment of Hanguang 800 NPUs.

<table>
<thead>
<tr>
<th>Business</th>
<th>Major App. Type</th>
<th>GPU Server</th>
<th>NPU Server</th>
<th>Deployment %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV-1</td>
<td>Image/Video</td>
<td>2x T4</td>
<td>2x 3-core</td>
<td>8.2%</td>
</tr>
<tr>
<td>CV-2</td>
<td>Image/Video</td>
<td>8x T4</td>
<td>2x 3-core</td>
<td>28.5%</td>
</tr>
<tr>
<td>CV-3</td>
<td>Image/Video</td>
<td>8x T4</td>
<td>2x 3-core</td>
<td>12.5%</td>
</tr>
<tr>
<td>CV-4</td>
<td>Image/Video</td>
<td>4x T4</td>
<td>4x 2-core</td>
<td>12.5%</td>
</tr>
<tr>
<td>RM-1</td>
<td>Recommender</td>
<td>1x T4</td>
<td>1x 4-core</td>
<td>12.5%</td>
</tr>
<tr>
<td>RM-2</td>
<td>Recommender</td>
<td>1x T4</td>
<td>1x 4-core</td>
<td>25.1%</td>
</tr>
</tbody>
</table>
Summary

- Hanguang 800 NPU takes lead in AI inference performance.
- Fully integrated with X-Dragon compute platform, Hanguang instances provide both performance of bare-metal and elasticity of cloud infrastructure.
- Being deployed to cloud customers, Hanguang estimates to lower system TCO by ~50% for various business applications.
- Stay tuned for more Hanguang powered Alibaba cloud services.
上古三剑一曰含光，视不可见，运之不知其所触，泯然无际，经物而物不觉。
Thank You!